

# UBL - AN UNIVERSAL BUSINESS LANGUAGE FOR XML

*Associate Professor Ph.D. Florin Mihai  
Asistant Ph.d. Student Ofelia Aleca  
Asistant Ph.D. Student Cristina Tartavulea  
Academy of Economic Studies, Bucharest*

**Abstract:** *Diversity of the application and software platform, same as necessity of rapid and efficiently communication between organizational software, have determined the development of a new specific language such as XML with particular subsets for specific areas, such as UBL. With UBL patterns there can delivered and received messages between business partners without fax and paper support. UBL define a free library of electronically standard XML documents such as payment documents or invoices. The purpose of this article is to present the UBL language and the way to adapt and implement of Romanian business.*

## **Introduction**

UBL documents are XML files with a standard structure based on the ebXML Core Components (ISO 15000-5) technical structure. UBL language contain 31 documents which describes extended scenarios and general transportation processes. The governments of Denmark, Norway, Sweden, England, Finland, Island, Singapore, Hong Kong and SUA financed the development of this scheme.

### **Figure 1 WEB publishing , WEB commerce parallel**

**HTTP + HTML = Web Publishing**

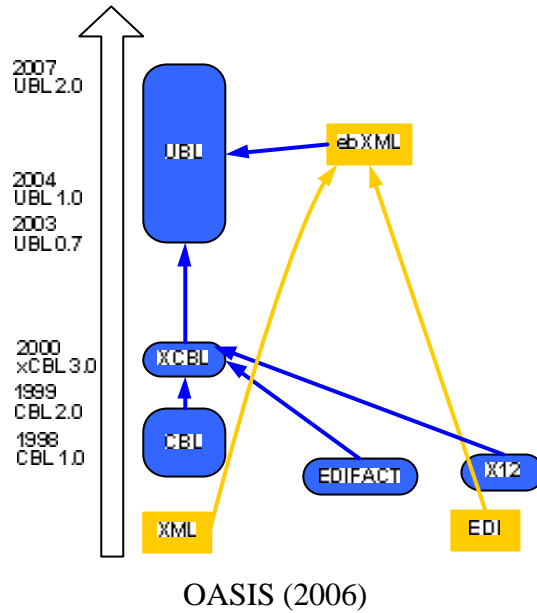
**ebXML + UBL = Web Commerce**

(Source: Tim McGrath)

## **1. History of UBL**

The ANSI institute has developed in 1979 the X12EDI standard; in about the same period it used the UN/ECE (*United Nations Economic Commission*) as EDIFACT standard which becomes an ISO standard in 1987. In the next years from EDIFACT, X12 and CBL was born XCBL which follows UBL. In December 2007 UBL becomes an international standard promoted by OASIS. UBL is based on ebXML Core Component library, which also followed the EDI standard

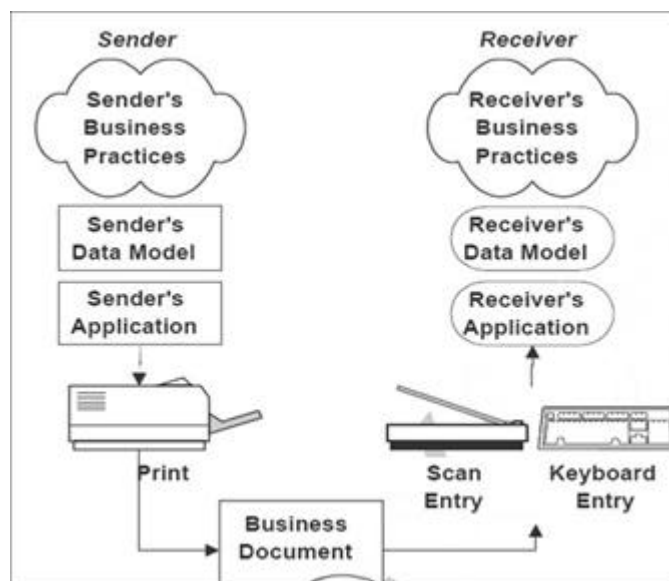
**Figure 2 UBL genesis**



## 2. UBL structure

When we want to realize a command in classic systems (on-line and desktop), the buyer provides the needed information for registration of the command, the salesman introduces data in the application and, based on the existing template, the application creates and print the order formulary. The following step is the documents exchange which can be done electronically (through fax or scanning and email) or classically through courier or direct. The receiver gets the ordered products and the corresponding documents which he must register in his application. The registration can be done manually or by scanning the document and recognizing the key fields.

**Figure 3 Documents flux in classical systems**

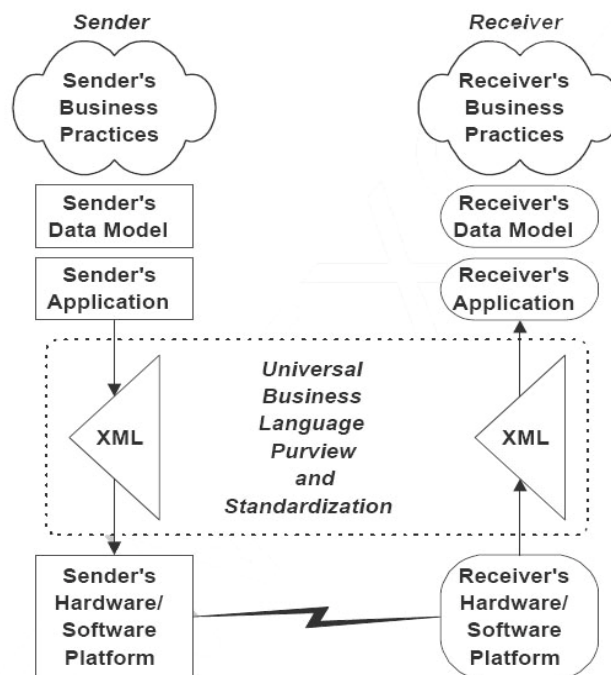


Source McGrath T. 2005

The disadvantages of this system are first the redundancy of the information's, the manifold introductions of data in different systems which are incompatible as structure and rolling platform, the time lost between sending and recovering data, eventual costs due to distance. Among the advantages of this system, we can mention the information security, the knowledge of procedures by employees.

The UBL language provides a easier way to transmit documents between sender and receiver. The documents are not transmitted on paper support, but on electronic support as XML files which has a standard structure recognized by any (all) applications. This way there will be no costs for the paper documents transport and transfer, and there will be no longer the need to introduce data into more information systems.

**Figure 4 Documents flux with UBL**



Source McGrath T. 2005

UBL language allows software application to communicate, giving a common language for communication. UBL provides a framework for communities of users and for trading partners to declare controlled vocabularies (e.g. code lists, identifiers) in use.

UBL 2.0 it was adopted as OASIS standard in December 2007 becoming in this way the UN/CEFACT specification. Being the sixth generation of XML standards, UBL 2.0 includes a complete set of libraries with 31 documents which can be grouped as follows:

- **Presale:** Request for quotation, Quotation, Catalog request, Catalog, Catalog deletion, Catalog item update, Catalog pricing update.
- **Ordering:** Order, Order response, Simple order response, Order cancellation, Order change.
- **Delivery:** Bill of lading, Certificate of origin, Forwarding instructions, Packing list, Transportation status, Way bill, Receipt advice, Dispatch advice
- **Invoicing:** Reminder, Invoice, Self billed invoice, Credit note, Self billed credit note, Debit note, Self billed debit note, Statement
- **Payment:** Remittance advice
- **Other:** Application response, Attached document

The use of UBL language improves the way to transmit documents, the main advantages being :

- The use of XML offers the independence of platform and software application.
- On line transmission of documents is more rapid and chipper.
- Standardization involves chipper software tools.
- Free tools are available to work with the documents

The purpose of UBL is not to redefine business procedure. It is addressed to economical information which respects a certain format and contents standard for exchanges between business partners. The use of UBL does not involve the modification of the way the information is stored, the documents exchange being independent of any internal schema of application. The UBL models can be extents and personalize with supplementary information non standard. Because it is basses on specific of every commercial transaction, UBL defines only the documents' structure and vocabulary for the exchange made through the W3C Schema (XSD) and of the labels and information encapsulated into it.

UBL disadvantages are (McGrath, 2005):

- Components name don't describe all the process to which they refer
- XML is not self-describing
- Modelers will often choose different names for the same component.
- Different document samples can lead to incompatible models.
- All model expressions have technological limitations.

UBL is projected to offer support in commercial communication between applications. UBL schemes are modularize and extensible. Like the first standard implemented by ebXML, UBL library is based on elements specific to conceptual models known as entities (Business Information Entities). These entities are integrated into specific documents such as Order and Invoice. The resulted documents are transformed using specific rules and notation into W3C XSD schema syntaxes. Because UBL documents have a W3C XSD schema, they can be easy understood by any application. (OASIS, 2006)

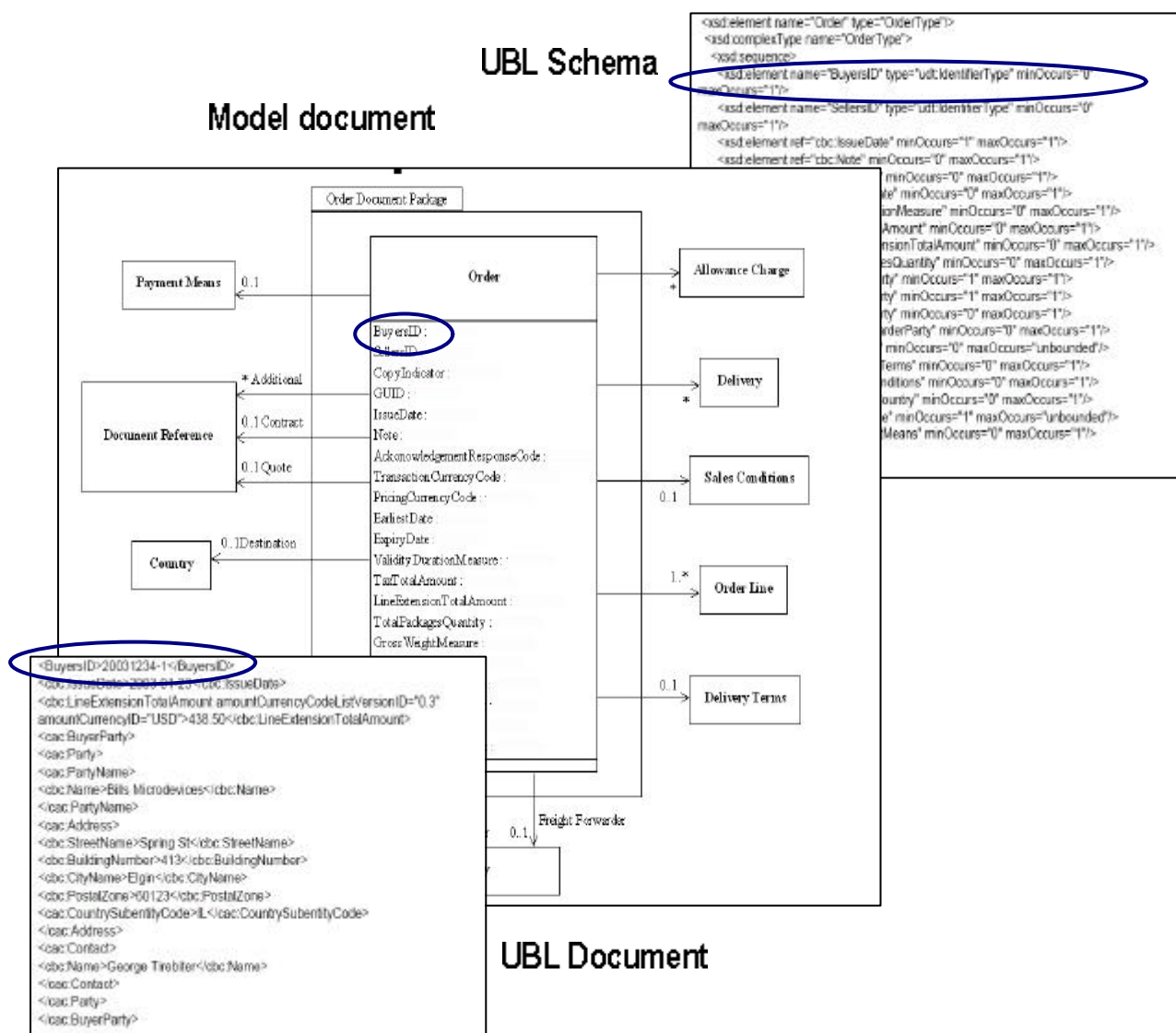
### **3. UBL in Romania**

The Romanian management applications environment is not a standardized one by the inter-applications communication point of view. At this moment there are a lot of integrated or stand-alone applications which offer a very small support or even none for information communication.

Even if it could look irrelevant, the simple web search in 7 October 2007 of the term "UBL" using Google provided for Romanian language only two links with information. This gives us a clue about the use of this standard. Considering the adoption in Romania of i2010 strategy from Lisbon for digitizing the economical activity of UE, the adoption of UBL standard becomes a necessity. The adoption of UBL for the standardization of information exchange and the development of on-line exchange of documents and information must, firs of all, be done through the adaptation of the documents included in the standard at the specific situation from Romania.

The paper documents represent about 60% of the total information from Romanian organizations, the intention for the next years being to reduce this percent to a half. The alignment of Romania to the European standards generates the accelerated increase of the Romanian market of applications for documents management, imposing the standardization of the information used in documents exchange.

Figure 5 UBL Conceptual model



Source OASIS (2006)

The Internet evolution has created many business opportunities. The virtual shops have developed a lot in the latest years. There can be bought on the internet various products, even certain services, existing the possibility to make payments and consult credits. To buy a product from a virtual shop you just have to choose it and fill your personal data in an order. The orders will be taken by the seller's application and will be printed as an invoice that will be send on paper together with the product. The data from the invoice will be manually filled by an employee in the client application, the invoice being scanned and eventually archived. The same information regarding this transaction has been processed by several persons, each working on different applications. The standardization of the communication language between applications would allow the transfer of the invoice on electronic form (UBL Document) from the supplier to the client and the automatically transfer of it in the client application without being necessarily to introduce it manually.

The UBL models library contain several types of invoice models as the traditional invoicing, the invoicing in the case of product return and the self-invoicing (when the client issue an invoice for himself, in the name and for the supplier, and sends to this one a copy). To each document UBL contains the description of the business process represented with UML activity diagram, the

conceptual data model represented with the UML class diagram, and the UBL scheme represented using an XSD syntax scheme. The adaptation of UBL standard for the situation in Romania presumes the adaptation of the components to specific situations. The main disadvantages of UBL standardization would be the obligation to introduce a large number of information in the documents and the amplification of security measures.

### **Bibliography**

1. McGrath T. (2005), Document Engineering with UBL: The Missing Pieces for Web Services
2. OASIS (2006), Universal Business Language v2.0